



# SQL

## SQL Basics

---

Seema Sirpal

Delhi University Computer Centre

# What is SQL?

- SQL (pronounced "ess-que-el") stands for Structured Query Language.
- SQL is used to communicate with a database.
- It is the standard language for relational database management systems.
- SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database.
- Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc.
- SQL is made up three sub-languages DDL, DML, DCL

# Table Basics

- A relational database system contains one or more objects called tables.
- The data or information for the database are stored in these tables.
- Tables are uniquely identified by their names and are comprised of columns and rows.
- Columns contain the column name, data type, and any other attributes for the column.
- Rows contain the records or data for the columns.

## Table Basics contd....

- Here is a sample table called "weather".
- City, state, high, and low are the columns. The rows contain the data for this table:

<b>Weather</b>			
<b>city</b>	<b>state</b>	<b>high</b>	<b>low</b>
Phoenix	Arizona	105	90
Tucson	Arizona	101	92
Flagstaff	Arizona	88	69
San Diego	California	77	60

# Selecting Data

- The **select** statement is used to query the database and retrieve selected data that match the criteria that you specify.
- Format of a simple select statement:

```
select "column1" [,"column2",etc] from  
"tablename" [where "condition"]; [] =  
optional
```

## Selecting Data contd...

- Conditional selections used in the **where** clause:

=Equal

>Greater than

<Less than

>=Greater than or equal

<=Less than or equal

<>Not equal to

LIKE

# Selecting Data contd...

- Examples :

```
select first, last, city from empinfo where first LIKE 'Er%';
```

```
select first, last from empinfo where last LIKE '%s';
```

```
select * from empinfo where first = 'Eric';
```

# Creating Tables

- The **create table** statement is used to create a new table.
- Format of a simple **create table** statement:

```
create table "tablename"  
("column1" "data type",  
"column2" "data type",  
"column3" "data type");
```



# Creating Tables contd...

- Most common Data types:

<code>char(size)</code>	Fixed-length character string. Size is specified in parenthesis. Max 255 bytes.
<code>varchar(size)</code>	Variable-length character string. Max size is specified in parenthesis.
<code>number(size)</code>	Number value with a max number of column digits specified in parenthesis.
<code>date</code>	Date value
<code>number(size,d)</code>	Number value with a maximum number of digits of "size" total, with a maximum number of "d" digits to the right of the decimal.

# Creating Tables contd...

- **Example:**

```
create table employee  
(first varchar(15),  
last varchar(20),  
age number(3),  
address varchar(30),  
city varchar(20),  
state varchar(20));
```

# Creating Tables contd...

- All SQL statements should end with a ";".
- The table and column names must start with a letter and can be followed by letters, numbers, or underscores.
- Table and column names not to exceed a total of 30 characters in length.
- Do not use any SQL reserved keywords as names for tables or column names (such as "select", "create", "insert", etc).

# What are constraints?

- A constraint is basically a rule associated with a column that the data entered into that column must follow.
- For example,

A "unique" constraint specifies that no two records can have the same value in a particular column. They must all be unique.

The other two most popular constraints are "not null" which specifies that a column can't be left blank,

and "primary key". A "primary key" constraint defines a unique identification of each record (or row) in a table.

## Constraints contd...

- Format of create table if you were to use optional constraints:

```
create table "tablename"  
("column1" "data type" [constraint],  
"column2" "data type" [constraint],  
"column3" "data type" [constraint]);
```

[ ] = optional

# Inserting into a Table

- The **insert** statement is used to insert or add a row of data into the table.

```
insert into "tablename"  
(first_column,...last_column) values  
(first_value,...last_value);
```

Example :

```
insert into employee  
(first, last, age, address, city, state)  
values ('Luke', 'Duke', 45, '2130 Boars Nest', 'Hazard  
Co', 'Georgia');
```

Strings should be enclosed in single quotes, and numbers should not.

# Updating Records

- The **update** statement is used to update or change records that match a specified criteria.

```
update "tablename"  
set "columnname" = "newvalue" [,"nextcolumn" =  
"newvalue2"...] where "columnname" OPERATOR  
"value" [and|or "column" OPERATOR "value"];
```

[] = optional

# Updating Records contd....

- Examples:

```
update phone_book set area_code = 623 where  
prefix = 979;
```

```
update phone_book set last_name = 'Smith',  
prefix=555, suffix=9292 where last_name = 'Jones';
```

```
update employee set age = age+1 where  
first_name='Mary' and last_name='Williams';
```



# Deleting Records

- The **delete** statement is used to delete records or rows from the table.

```
delete from "tablename"  
where "columnname" OPERATOR "value"  
[and/or "column" OPERATOR "value"];
```

[ ] = optional

## Deleting Records contd...

- **Examples:**

delete from employee;

**Note:** if you leave off the where clause, **all records will be deleted!**

delete from employee where lastname = 'May'; delete from employee where firstname = 'Mike' or firstname = 'Eric';